



2022.01.21 ~ 2022.02.21

ZOOZOO CLUB

Infinity

1.1 프로젝트 목적

- 유기동물 및 반려동물, 동물병원 관련 정보 게시판 제공
- 수업 내용 개념 재확립 및 계층형 게시판 구조 설계

1.2 Target

- 유기동물 및 반려동물 입양을 원하는 고객
- 그 외 동물 관련 정보를 원하는 고객
(동물병원정보, 입양후기 등)

1.3 개발환경

- 운영체제 : Windows10
- Front-end : HTML, CSS, JavaScript, Bootstrap
- Back-end : JDK 11, Spring Framework
- Database : MySQL
- Server : Tomcat
- Version Control : Git

1 개요

1.4 프로젝트 기간

2022.01.21 ~ 2022.02.21 (약 5주 - 총 32일)

[illegible]

1.5 개발인원, 담당역할

강보균 (조장)

- 분양동물 게시판, Main page, 회원가입, 로그인
- 프로젝트 설계, 팀원 역할분담, 팀원 서포트, Github Readme 작성

장용범 (조원)

- 유기동물 게시판, Main page, 비밀번호 찾기, Mypage
- Github Readme 작성

장희동 (조원)

- 자유 게시판, 후기 게시판, 아이디 찾기

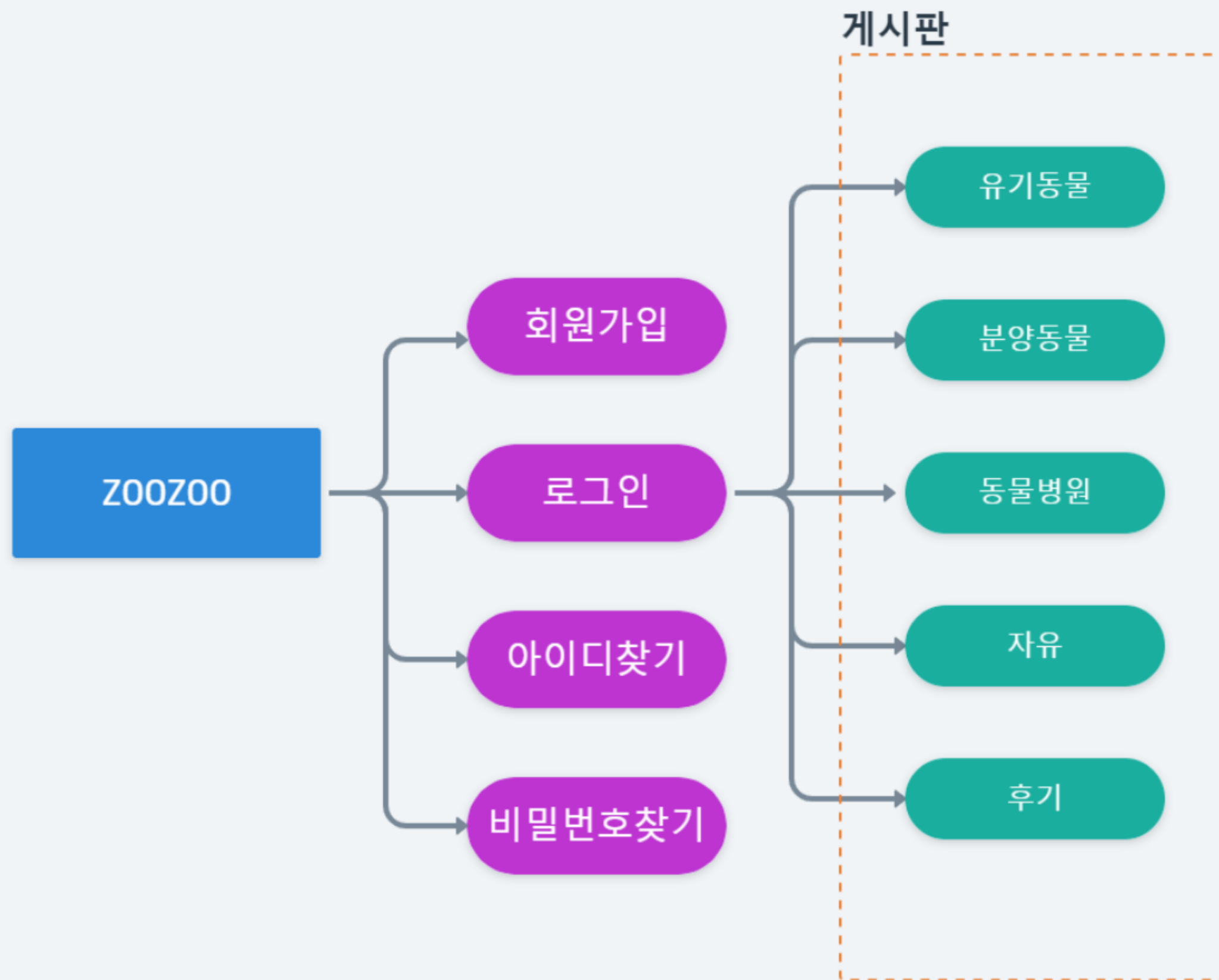
최병호 (조원)

- 동물병원 게시판, Main page

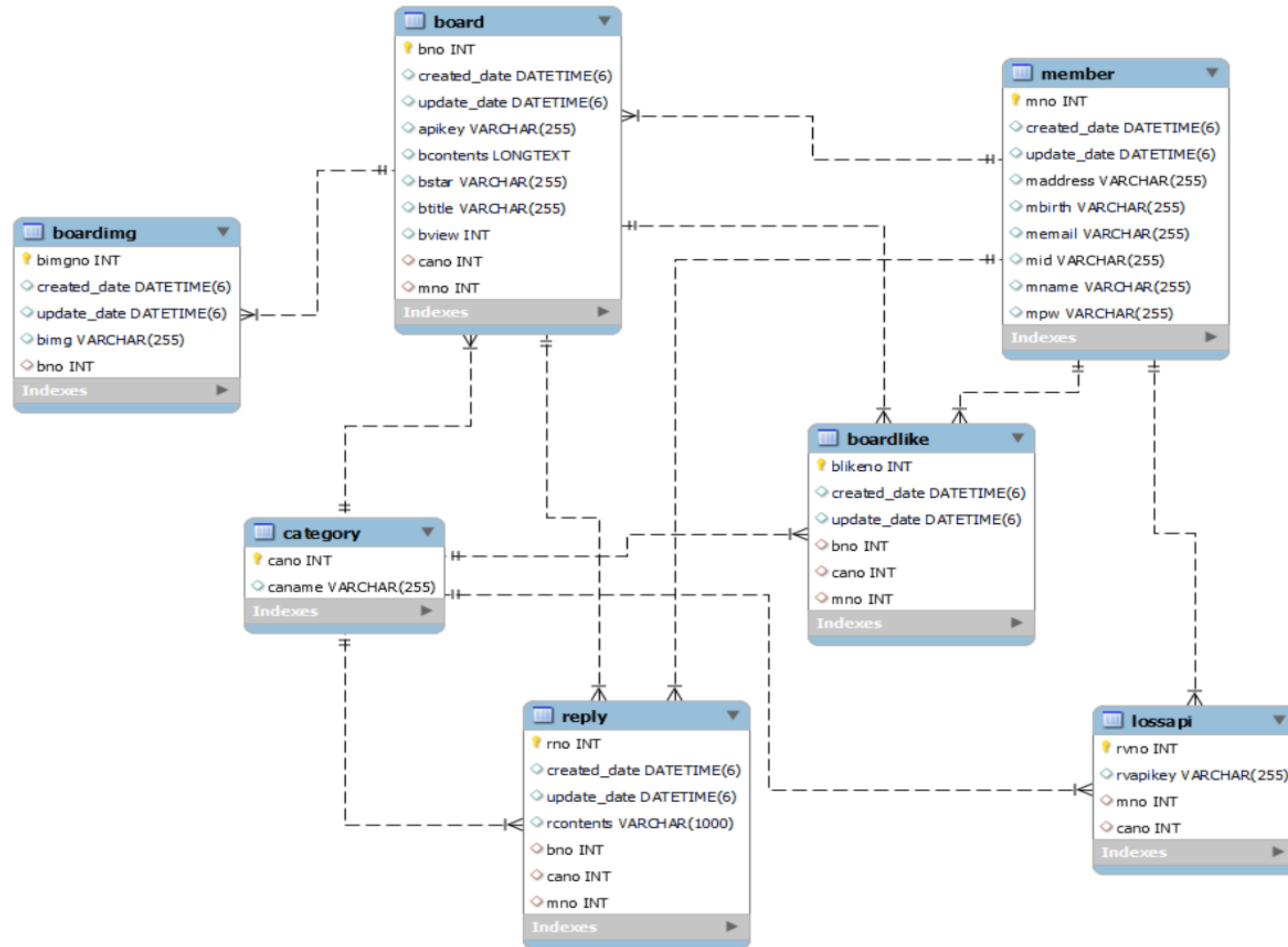
공통

- 주제 선정, UI설계, DB설계

2 화면구조도



3 ERD



4 프로젝트 기술 상세

1. Open API 호출 - XML

```
public ArrayList<LossDTO> totlosslist() {  
  
    ArrayList<LossDTO> lossDTOS = new ArrayList<>();  
  
    try {  
        // max page = 110  
        String urlStr = "https://openapi.gg.go.kr/AbdAnimalProtect?KEY=f116bb9347d04a38a639e01395505d21&pIndex=1&pSize=1000";  
        // Instantiate the Factory  
        DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();  
        // optional, but recommended  
        // process XML securely, avoid attacks like XML External Entities (XXE)  
        dbf.setFeature(XMLConstants.FEATURE_SECURE_PROCESSING, value: true);  
  
        // parse XML file  
        DocumentBuilder db = dbf.newDocumentBuilder();  
        Document doc = db.parse(urlStr);  
  
        // optional, but recommended  
        doc.getDocumentElement().normalize();  
  
        // get <staff>  
        NodeList list = doc.getElementsByTagName("row");
```

→ 1 →

```
for (int temp = 0; temp < list.getLength(); temp++) {  
    Node node = list.item(temp);  
    Element element = (Element) node;  
  
    if (node.getNodeType() == Node.ELEMENT_NODE) {  
  
        // get text  
        String SIGUN_CD = element.getElementsByTagName("SIGUN_CD").item(0).getTextContent();  
        String SIGUN_NM = element.getElementsByTagName("SIGUN_NM").item(0).getTextContent();  
        String ABDM_IDNTFY_NO = element.getElementsByTagName("ABDM_IDNTFY_NO").item(0).getTextContent();  
        String THUMB_IMAGE_COURS = element.getElementsByTagName("THUMB_IMAGE_COURS").item(0).getTextContent();  
        // 접수일자 - 삽입  
        String RECEPT_DE = element.getElementsByTagName("RECEPT_DE").item(0).getTextContent();  
        StringBuffer newRECEPT_DE = null;  
        StringBuffer buffer = new StringBuffer(RECEPT_DE);  
        buffer.insert(offset: 4, str: "-");  
        buffer.insert(offset: 7, str: "-");  
        newRECEPT_DE = buffer;  
        String DISCVRY_PLC_INFO = element.getElementsByTagName("DISCVRY_PLC_INFO").item(0).getTextContent();  
        String STATE_NM = element.getElementsByTagName("STATE_NM").item(0).getTextContent();  
        String SPECIES_NM = element.getElementsByTagName("SPECIES_NM").item(0).getTextContent();  
        SPECIES_NM = SPECIES_NM.replace(target: "[", replacement: "");  
        String result = SPECIES_NM.split(regex: "[ ]")[0];  
        String details = null;  
        try {  
            details = SPECIES_NM.split(regex: "[ ]")[1];
```

← 2 ←

```
LossDTO lossDTO = LossDTO.builder()  
    .SIGUN_CD(SIGUN_CD).SIGUN_NM(SIGUN_NM).ABDM_IDNTFY_NO(ABDM_IDNTFY_NO).THUMB_IMAGE_COURS(THUMB_IMAGE_COURS)  
    .RECEPT_DE(RECEPT_DE).newRECEPT_DE(newRECEPT_DE).DISCVRY_PLC_INFO(DISCVRY_PLC_INFO).SPECIES_NM(result)  
    .details(details).COLOR_NM(COLOR_NM).AGE_INFO(AGE_INFO).BDWGH_INFO(BDWGH_INFO).PBLANC_IDNTFY_NO(PBLANC_IDNTFY_NO)  
    .PBLANC_BEGIN_DE(PBLANC_BEGIN_DE).newPBLANC_BEGIN_DE(newPBLANC_BEGIN_DE).PBLANC_END_DE(PBLANC_END_DE)  
    .newPBLANC_END_DE(newPBLANC_END_DE).IMAGE_COURS(IMAGE_COURS).STATE_NM(STATE_NM).SEX_NM(sex).NEUT_YN(NEUT_YN)  
    .SFETR_INFO(SFETR_INFO).SHTER_NM(SHTER_NM).SHTER_TELNO(SHTER_TELNO).PROTECT_PLC(PROTECT_PLC).JURISD_INST_NM(JURISD_INST_NM)  
    .CHRGPSN_NM(CHRGPSN_NM).CHRGPSN_CONTCT_NO(CHRGPSN_CONTCT_NO).PARTCLR_MATR(PARTCLR_MATR).REFINE_LOTNO_ADDR(REFINE_LOTNO_ADDR)  
    .REFINE_ROADNM_ADDR(REFINE_ROADNM_ADDR).city(city).REFINE_ZIP_CD(REFINE_ZIP_CD).REFINE_WGS84_LOGT(REFINE_WGS84_LOGT)  
    .REFINE_WGS84_LAT(REFINE_WGS84_LAT)  
    .build();  
    lossDTOS.add(lossDTO);  
}  
return lossDTOS;
```


4 프로젝트 기술 상세

2. Open API 호출 - JSON

```
public int getapitotal(){
    try {
        String asd2 [] = new String[2];
        String test[] = new String[2];
        URL url = new URL( spec: "https://openapi.gg.go.kr/Animalhospt1?KEY=ccec045fa339456eaefa46cb9c828bc8&Type=json&pIndex=1&pSize=1000");
        BufferedReader bf = new BufferedReader( new InputStreamReader( url.openStream() , charsetName: "UTF-8") );
        String result = bf.readLine();
        JSONParser jsonParser = new JSONParser();
        JSONObject jsonObject = (JSONObject)jsonParser.parse(result);
        JSONArray jsonArray = (JSONArray)jsonObject.get("Animalhospt1");
        JSONObject jsonObject3 = (JSONObject) jsonArray.get(0);
        JSONArray jsonArray2 = (JSONArray)jsonObject3.get("head");
        String asd = jsonArray2.get(0).toString();
        asd2 = asd.split( regex: ":" );
        test = asd2[1].split( regex: "}" );
        int total = Integer.parseInt(test[0]);
        return total;
    }catch(Exception e){
        return 0;
    }
}
```


4 프로젝트 기술 상세

3. 다중옵션 검색

유기

분양

병원

자유

후기

상태

전체

성별

전체


품종

전체

보호지역(시군구)

전체

검색



믹스견

접수일자

2022-02-19

품종

개

보호지역

오산시

나이

2018(년생)

성별


암컷

상태

보호중

공고기간

2022-02-19 ~ 2022-03-02



믹스견

접수일자

2022-02-19

품종

개

보호지역

남양주시

나이

2020(년생)

성별


수컷

상태

보호중

공고기간

2022-02-19 ~ 2022-03-02



믹스견

접수일자

2022-02-19

품종

개

보호지역

남양주시

나이

2020(년생)

성별


수컷

상태

보호중

공고기간

2022-02-19 ~ 2022-03-02



믹스견

접수일자

2022-02-19

품종

개

보호지역

남양주시

나이

2016(년생)

성별

암컷

상태

보호중

공고기간

2022-02-19 ~ 2022-03-02

4 프로젝트 기술 상세 - 유기동물 게시판

3. 다중옵션 검색

- html의 각 select box에서 요청받은 value값을 변수로 검색 경우의 수 조건 설정

HTML

```
<form action="/LossBoardlist" method="get">
  <div class="row">
    <div class="col-md-2 d-flex"> <!-- d-flex-->
      <label style="width:100px; font-size:18px; color:white;"><strong>상태</strong></label>
      <select name="state" class="form-select">
        <option value="total" selected="selected">전체</option>
        <option value="보호중">보호중</option>
        <option value="종료">종료</option>
      </select>
    </div>
    <div class="col-md-2 d-flex"> <!-- d-flex-->
      <label style="width:100px; font-size:18px; color:white;"><strong>성별</strong></label>
      <select name="sex" class="form-select">
        <option value="total" selected="selected">전체</option>
        <option value="암컷">암컷</option>
        <option value="수컷">수컷</option>
        <option value="미상">미상</option>
      </select>
    </div>
    <div class="col-md-2 d-flex">
      <label style="width:100px; font-size:18px; color:white;"><strong>품종</strong></label>
      <select name="kind" class="form-select">
        <option value="total" selected="selected">전체</option>
        <option value="개">개</option>
        <option value="고양이">고양이</option>
        <option value="기타종류">기타</option>
      </select>
    </div>
  </div>
</form>
```

Controller

```
@GetMapping("/LossBoardlist")
public String goToLossBoardlist(Model model, @RequestParam(defaultValue = "1") int page) {

    String sex = request.getParameter( name: "sex"); // 성별
    String kind = request.getParameter( name: "kind"); // 종족
    String city = request.getParameter( name: "city"); // 사군구
    String state = request.getParameter( name: "state"); // 상태
    HttpSession session = request.getSession();

    if (sex != null || kind != null || city != null || state != null) {
        session.setAttribute( name: "sex", sex);
        session.setAttribute( name: "kind", kind);
        session.setAttribute( name: "city", city);
        session.setAttribute( name: "state", state);
    } else {
        sex = (String) session.getAttribute( name: "sex");
        kind = (String) session.getAttribute( name: "kind");
        city = (String) session.getAttribute( name: "city");
        state = (String) session.getAttribute( name: "state");
    }

    ArrayList<LossDTO> parses = lossService.losslist(sex, kind, city, state); // 필터링 게시물
    ArrayList<LossDTO> parsesPage = lossService.parsenum(parses, page); // 페이지
    Pagination1 pagination = new Pagination1(parses.size(), page);
}
```

Service

```
public ArrayList<LossDTO> losslist(String sex, String kind, String city, String state) {
    ArrayList<LossDTO> parses = new ArrayList<>();
    try {
        if (sex != null && kind != null && city != null && state != null) {
            parses = lossRepository.findAllBySexAndKindAndCityAndState(sex, kind, city, state);
        } else if (sex != null && kind != null && city != null) {
            parses = lossRepository.findAllBySexAndKindAndCity(sex, kind, city);
        } else if (sex != null && kind != null) {
            parses = lossRepository.findAllBySexAndKind(sex, kind);
        } else if (sex != null) {
            parses = lossRepository.findAllBySex(sex);
        } else if (kind != null) {
            parses = lossRepository.findAllByKind(kind);
        } else if (city != null) {
            parses = lossRepository.findAllByCity(city);
        } else if (state != null) {
            parses = lossRepository.findAllByState(state);
        } else {
            parses = lossRepository.findAll();
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return parses;
}
```

Controller

```
model.addAttribute( attributeName: "parsesPage", parsesPage);
model.addAttribute( attributeName: "pagination", pagination);
return "Board/Loss/LossBoardlist";
```

1. 모든 게시판에 무한 계층형 댓글 기능 추가

- 현재 일부 페이지만 1depth까지 대댓글 기능 구현

강보균

API가 주가되는 게시판을 만들면서 DB와 Entity를 사용하지 않아 따라오는 어려움이 있었습니다. 데이터를 쉽게 가공하지 못하고 가공을 하려면 나름의 알고리즘을 a부터 z까지 짜서 사용했어야 해서 시간이 조금 더 걸렸었지만 수업시간에 진행하지 않았던 엔티티를 이용하지 않는 페이징, 시간출력 포맷, 대댓글 등 까지 생각해보고 설계하여 코드를 짜보는 귀중한 경험을 하고 한단계 발전하는 시간이 됐습니다.

장용범

Open API의 데이터를 사용하면서 Entity를 사용하지 않는 부분에 대해 어려움이 있었습니다. 특히 다른 팀원들과 달리 API 타입이 xml이고 수업시간에도 다뤄보지 않아서 많은 어려움이 있었습니다. Open API 가공, Pagination 등 직접 알고리즘을 구축하고 기능을 구현하며 Open API에 대해 좀 더 공부할 수 있는 시간을 가질 수 있었습니다. 또한 계층형 게시판 구조로 설계하고 게시글과 댓글의 CRUD를 구현하면서 게시판에 대해서도 공부할 수 있는 귀중한 시간이었습니다.

장희동

계층형 DB를 접하면서, 이제까지 배웠던 것들보다 더 압축된 DB 디자인을 따르는 것이 처음이라 막막했습니다. 모르는 부분은 팀원들에게 물어보면서 최대한 제 스스로 해결하려고 노력했습니다. 또한 대댓글 기능을 구현하면서 실제 상용화되는 사이트들은 더욱이 견고하고 촘촘한 DB구조를 사용하고 있다는 점을 느끼게 되었습니다.

최병호

동물 병원 게시판을 진행하면서 API를 활용하여 데이터를 가공해서 써야 하는 경우가 대다수 였는데 직접 알고리즘을 짜서 페이지를 출력하는 과정을 통해서 게시판과 페이징에 대해 다시 되돌아보는 시간이 되었습니다.

또한 이번 프로젝트를 통해서 설계의 중요성을 다시 한번 돌아보게 되었는데 로직을 돌리는 과정이 많다 보니 페이지의 출력 부분이 느려지는 부분을 체감하여 항상 코드가 되기만 하는 과정보다는 어떻게 하면 코드를 조금 더 효율적으로 작성할 수 있을지에 대한 생각을 해보게 되었습니다.



2022.01.21 ~ 2022.02.21

Thinking Infinity

THANK YOU.